Subject: Re: Can I have multiple threads in IDL?
Posted by Peter Mason on Sun, 08 Aug 2004 22:16:12 GMT
View Forum Message <> Reply to Message

Marc Reinig wrote:
> We're currently running a very large computational program on a
> multiprocessor machine. When the FFT portion runs, the CPU load on
> all 8 processors spikes, so IDL is making use of them, life is good.
>
> However, we have another portion of the code that dose a massive
> nested loop on some arrays, but the operation is carried out on a
> very small subset of the array and IDL uses only a single processor.
> It could be easily broken into threads if we were writing in C to
> allow the OS (Windows Server 2003) to run the threads simultaneously.
>
> We would like to break this processing into multiple threads in IDL
> so that the task can be run on the other 7 processors in a similar
> manner. Is there a way to create and run a thread in IDL?

There is nothing about IDL itself that will stop you from writing threaded external code. If you can write a CALL_EXTERNAL or DLM module and you can write threaded code in C then you're set. OTOH, there is nothing in IDL that will assist you here. IDL's internal threading system is not available to external code. (Last I looked anyway.) You have to figure out how best to divide up the work into chunks and you have to do all your own thread creation and management in your C module. IMO, the first part is usually the hard part. The second part is usually pretty straightforward for a dedicated computational module; trivial, in fact, if you have designed things so that your threads "keep off one another's turf".

Peter Mason

Subject: Re: Can I have multiple threads in IDL?
Posted by Marc Reinig on Mon, 09 Aug 2004 14:11:08 GMT
View Forum Message <> Reply to Message

Peter,

Thanks for the response.

I'm fairly new to IDL, so ... ;=)

I know I can call IDL from a C program and use some of its functions. Can I use that to help me? I am unsure of the limitations.

I have a complex function in IDL. IDL won't break up the computations into multiple threads for me the way it does for FFTs.

So I was thinking of a different way to solve the problem without rewriting the entire function in C.

Could I call a function in a DLM and pass it the total parameters. Have the DLM break it into 8 pieces, create 8 threads, and each thread would call my IDL function with it's piece, the original function would return with the result of the original call?

Could that work?

Thanks in advance,

Marc Reinig
UCO/Lick
LAO

"Peter Mason" <drone@spam.com> wrote in message news:MCxRc.228$aA.9451@news.optus.net.au...
> Marc Reinig wrote:
>> We're currently running a very large computational program on a
>> multiprocessor machine. When the FFT portion runs, the CPU load on
>> all 8 processors spikes, so IDL is making use of them, life is good.
>>
>> However, we have another portion of the code that dose a massive
>> nested loop on some arrays, but the operation is carried out on a
>> very small subset of the array and IDL uses only a single processor.
>> It could be easily broken into threads if we were writing in C to
>> allow the OS (Windows Server 2003) to run the threads simultaneously.
>>
>> We would like to break this processing into multiple threads in IDL
>> so that the task can be run on the other 7 processors in a similar
>> manner. Is there a way to create and run a thread in IDL?
>
> There is nothing about IDL itself that will stop you from writing threaded
> external code. If you can write a CALL_EXTERNAL or DLM module and you can
> write threaded code in C then you're set. OTOH, there is nothing in IDL
> that will assist you here. IDL's internal threading system is not
> available to external code. (Last I looked anyway.) You have to figure
> out how best to divide up the work into chunks and you have to do all your
> own thread creation and management in your C module. IMO, the first part
> is usually the hard part. The second part is usually pretty
> straightforward for a dedicated computational module; trivial, in fact, if
> you have designed things so that your threads "keep off one another's turf".
>

> Peter Mason
>
>

Subject: Re: Can I have multiple threads in IDL?
Posted by Peter Mason on Mon, 09 Aug 2004 21:56:42 GMT
View Forum Message <> Reply to Message

Marc Reinig wrote:
> Peter,
>
> Thanks for the response.
>
> I'm fairly new to IDL, so ... ;=)
>
> I know I can call IDL from a C program and use some of its functions.
> Can I use that to help me? I am unsure of the limitations.
>
> I have a complex function in IDL. IDL won't break up the computations
> into multiple threads for me the way it does for FFTs.
>
> So I was thinking of a different way to solve the problem without
> rewriting the entire function in C.
>
> Could I call a function in a DLM and pass it the total parameters.
> Have the DLM break it into 8 pieces, create 8 threads, and each
> thread would call my IDL function with it's piece, the original
> function would return with the result of the original call?

Hi Marc,

I'm out of my depth here. Last time I used anything like callable IDL was... an embarrassingly long time ago. (Those days it was called RPC.)
I generally go the other way, using IDL as the "host" application.
It's possible that the approach you outlined above might work but there are alarm bells going off in my mind: I have to wonder whether an IDL function call is thread-safe. I know that you can do recursive function calls within IDL (i.e., there is re-entrancy at that level), but there might be more involved in calling an IDL function from several concurrent external threads.
Hopefully, someone who reads this group has tried it and might post an answer. Failing that, I can only suggest that you try it out - perhaps with a simple task rather than the full-blown one.
Wouldn't it be nice if RSI exposed a threading mechanism in IDL? Even one with constraints? (Nudges RSI :-)

Cheers
Peter Mason